

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**

**ЗАТВЕРДЖУЮ**

Проректор з науково-педагогічної  
та навчальної роботи

\_\_\_\_\_ Світлана ЛУТКОВСЬКА

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
Тестування програмного забезпечення**

**Рівень вищої освіти Перший (бакалаврський)**

**Галузь знань 12 Інформаційні технології**

**Спеціальність 122 Комп'ютерні науки**

**Освітньо-професійна програма Комп'ютерні науки**

Вінниця 2024

Робоча програма навчальної дисципліни «Тестування програмного забезпечення». Рівень вищої освіти перший (бакалаврський), галузь знань 12 Інформаційні технології, спеціальність 122 Комп'ютерні науки, освітньо-професійна програма «Комп'ютерні науки». 2024, 24 с.

**Розробник:**

Суприган В. А., кандидат технічних наук, старший викладач кафедри комп'ютерних наук та цифрової економіки

**Викладачі:**

Суприган В. А., кандидат технічних наук, старший викладач кафедри комп'ютерних наук та цифрової економіки

Робочу програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та цифрової економіки

Протокол від «29» липня 2024 року № 1

Завідувач кафедри \_\_\_\_\_ Світлана КОЛЯДЕНКО

Робочу програму розглянуто і затверджено на засіданні навчально-методичної комісії Навчально-наукового інституту економіки та управління

Протокол від «30» липня 2024 року № 1

Голова навчально-методичної комісії  
ННІ економіки та управління \_\_\_\_\_ Лідія ФЕДОРИШИНА

Робочу програму розглянуто і затверджено на засіданні навчально-методичної комісії університету

Протокол від «31» липня 2024 року № 1

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітньо-професійна програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5	12 Інформаційні технології	Обов'язкова	
Атестацій – 2		<b>Рік підготовки:</b>	
Загальна кількість годин – 150	<u>122 Комп'ютерні науки</u>	3-й	3-й
		<b>Семестр</b>	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 7	<u>Комп'ютерні науки</u>  Перший (бакалаврський)	5-й	5-й
		<b>Лекції</b>	
		26 год.	6 год.
		<b>Практичні, семінарські</b>	
		24 год.	4 год.
		<b>Самостійна робота</b>	
100 год.	140 год.		
		Вид контролю: іспит	

Програма навчальної дисципліни передбачає перезарахування кредитів освітніх компонентів, отриманих студентами, які навчались за програмою академічної мобільності, неформальної та інформальної освіти за наявності відповідних підтверджуючих документів.

## **2. Мета та завдання навчальної дисципліни**

Метою вивчення дисципліни є формування у здобувачів знань, вмінь та професійних навичок необхідних для забезпечення дотримання заданих критеріїв якості у процесі розбудови сучасного програмного забезпечення, дотримання стандартних та загальноприйнятих методології функціонування процесів у інформаційних технологіях на мікро- та макрорівнях у різних галузях господарювання, зокрема у АПК.

У підсумку, здобувачі мають володіти основними підходами до організації розробки методології тестування та проведення тестування програмних систем, створення модульних та інтеграційних тестів, формування критичного мислення для складання тест-кейсів. Організації тестування на різних фазах розробки програмної системи.

Завданням навчальної дисципліни є оволодіння основами тестування ІТ-проектів, систем та прикладних програм; набуття вмінь і методології у тестуванні ІТ-проектів в АПК із застосуванням інформаційних технологій; набуття теоретичних та практичних навичок, самостійного засвоєння теоретичних, методичних основ забезпечення відповідності якості вимогам; набуття знань як вихідної основи для формування навичок і вмінь дослідження практичних проблем у цій сфері.

Після вивчення дисципліни студенти будуть вміти аналізувати та створювати вимоги та перевіряти якісні показники у прикладних програм та інформаційних систем. Знати етапи розробки та тестування програмної системи. Вміти вибрати метрики оцінки якості системи. Розробляти набори тест-кейсів, модульні тести та вміти організувати процедуру тестування.

## **3. Компетентності та результати навчання**

У результаті вивчення навчальної дисципліни здобувач вищої освіти повинен володіти інтегральною, загальними та фаховими компетентностями, зокрема:

*Інтегральна компетентність (ІК):*

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

*загальні компетентності (ЗК):*

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

ЗК10. Здатність бути критичним і самокритичним.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

*спеціальні (фахові) компетентності (СК):*

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК6. Здатність до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризики.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

## **ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ ВІДПОВІДНО ОСВІТНЬОЇ ПРОГРАМИ**

РН09. Розробляти програмні моделі предметних середовищ, вибрати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

РН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

РН14\*. Володіти технічними та інструментальними засобами, для проектування та забезпечення функціонування комп'ютерних систем, мережних технологій, розробки архітектури комп'ютерних мереж, мати практичні навички їх технологічного обслуговування та експлуатації.

РН15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

Вивчення даної дисципліни формує у здобувачів вищої освіти соціальні навички (soft skills): комунікативність (реалізується через: метод роботи в парах та групах, робота з інформаційними джерелами), робота в команді (реалізується через: метод проектів), лідерські навички (реалізується через: робота в групах, метод проектів, метод самопрезентації).

#### **4. Передумови для вивчення дисципліни**

Навчальна дисципліна базується на вивченні таких дисциплін: «Програмування», «Інформаційні технології», «Дискретна математика», «Web-технології та Web-дизайн».

#### **5. Програма навчальної дисципліни**

##### **Атестація 1**

##### **Теоретико-методичне вивчення тестування програмного забезпечення**

###### **Тема 1. Вступ до тестування програмного забезпечення.**

Важливість тестування програмного забезпечення для забезпечення надійності. Роль тестування програмного забезпечення в підвищенні задоволеності користувачів. Реальні приклади збоїв програмного забезпечення через недостатнє тестування.

###### **Тема 2. Основи тестування.**

Розуміння етапу планування тесту. Розробка тестів і документація. Значення простежуваності в тестуванні.

###### **Тема 3. Види тестування.**

Поглиблене дослідження модульного тестування та його переваг. Стратегії і найкращі практики інтеграційного тестування. Роль тестування системи в забезпеченні повної функціональності.

###### **Тема 4. Автоматизація тестування за допомогою Selenium.**

Знайомство з Selenium WebDriver і його можливостями. Створення сценаріїв автоматизованих тестових випадків з Selenium. Обробка динамічних веб-елементів і стратегій очікування в Selenium.

###### **Тема 5. Методи розробки тестів.**

Поділ еквівалентності. Розбиття вхідних даних на класи еквівалентності. Аналіз граничних значень. Тестування на краях вхідних доменів. Тестування таблиці рішень. керування складними комбінаціями умов.

###### **Тема 6. Виконання тесту та звітність.**

Впровадження наборів тестів для регресійного тестування. Ефективне повідомлення про дефекти за допомогою систем відстеження проблем. Методи пріоритезації тестових випадків.

## Атестація 2

### Тестування складних систем

#### **Тема 7. Безперервне тестування в хмарних середовищах.**

Налаштування хмарної інфраструктури тестування. Безперервна інтеграція та конвеєрна інтеграція безперервного розгортання (CI/CD). Організація тестів у хмарних середовищах для масштабованості.

#### **Тема 8. Тестування безпеки та контейнери Docker.**

Виявлення типових вразливостей безпеки в контейнерних програмах. Інструменти та техніки для сканування образів Docker на наявність вразливостей. Написання тестів безпеки для контейнерних програм.

#### **Тема 9. Тестування продуктивності в Kubernetes.**

Планування та проектування тестів ефективності. Використання Kubernetes для створення реалістичних тестових середовищ. Аналіз та інтерпретація результатів тестування продуктивності.

#### **Тема 10. Перевірка зручності використання та взаємодії з користувачем.**

Проведення інтерв'ю та опитувань користувачів для тестування зручності використання. Оцінка доступності та відповідності стандартам «юзабіліті». Збір і аналіз відгуків користувачів для покращення UX.

#### **Тема 11. Тестування в середовищах розробки.**

Налаштування робочих процесів тестування середовища розробки. Найкращі методи перевірки коду для раннього виявлення дефектів. Стратегії впровадження та виконання тестів, орієнтованих на розробника.

#### **Тема 12. Тестування прийнятності користувача (UAT) у виробництві.**

Залучення кінцевих користувачів до планування та виконання UAT. Визначення загальних проблем UAT та їх вирішення. Забезпечення плавного переходу від UAT до виробництва.

#### **Тема 13. Сучасні технології постійного вдосконалення тестування.**

Впровадження циклів зворотного зв'язку для постійного вдосконалення. Використання показників тестування та показників ефективності. Заохочення культури навчання та адаптації в командах тестування.

## 6. Структура навчальної дисципліни

Теми	Кількість годин											
	Денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Атестація 1. Теоретико-методичне вивчення тестування програмного забезпечення</b>												
<b>Тема 1.</b> Вступ до тестування програмного забезпечення	14	2	2	-	-	10	19	2	2	-	-	14
<b>Тема 2.</b> Основи тестування	13	2	2	-	-	9	13	2	-	-	-	11
<b>Тема 3.</b> Види тестування	12	2	2	-	-	8	11	-	-	-	-	11
<b>Тема 4.</b> Автоматизація тестування за допомогою Selenium	12	2	2	-	-	8	11	-	-	-	-	11
<b>Тема 5.</b> Методи розробки тестів	12	2	2	-	-	8	11	-	-	-	-	11
<b>Тема 6.</b> Виконання тесту та звітність	12	2	2	-	-	8	11	-	-	-	-	11
<b>Разом</b>	<b>75</b>	<b>12</b>	<b>12</b>	-	-	<b>51</b>	<b>75</b>	<b>4</b>	<b>2</b>	-	-	<b>69</b>
<b>Атестація 2. Тестування складних систем</b>												
<b>Тема 7.</b> Безперервне тестування в хмарних середовищах	11	2	2	-	-	7	15	2	2	-	-	11
<b>Тема 8.</b> Тестування безпеки та контейнери Docker	11	2	2	-	-	7	10	-	-	-	-	10
<b>Тема 9.</b> Тестування продуктивності в Kubernetes	11	2	2	-	-	7	10	-	-	-	-	10
<b>Тема 10.</b> Перевірка зручності використання та взаємодії з користувачем	11	2	2	-	-	7	10	-	-	-	-	10
<b>Тема 11.</b> Тестування в середовищах розробки	11	2	2	-	-	7	10	-	-	-	-	10



1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Тема 12.</b> Тестування прийнятності користувача (UAT) у виробництві	11	2	2	-	-	7	10	-	-	-	-	10
<b>Тема 13.</b> Сучасні технології постійного вдосконалення тестування	9	2	-	-	-	7	10	-	-	-	-	10
<b>Разом</b>	<b>75</b>	<b>14</b>	<b>12</b>			<b>49</b>	<b>75</b>	2	2	-	-	<b>71</b>
<b>Усього годин</b>	<b>150</b>	<b>26</b>	<b>24</b>	-	-	<b>100</b>	<b>150</b>	<b>6</b>	<b>4</b>	-	-	<b>140</b>

## 7. Теми практичних занять

№ з/п	Назва теми	Кількість годин	
		денна	заочна
<b>Атестація 1. Теоретико-методичне вивчення тестування програмного забезпечення</b>			
1	Автоматичне налаштування тестового сценарію за допомогою модульного тестування Python	2	2
2	Модульне тестування за допомогою Python	2	-
3	Інтеграційне тестування з Python	2	-
4	Автоматизація тестування Selenium за допомогою Python	2	-
5	Тестування API за допомогою Python	2	
6	Тестування навантаження за допомогою Python із Locust	2	
<b>Атестація 2. Тестування складних систем</b>			
7	Інструменти тестування безпеки в Python	2	2
8	Ручне тестування ендпоінтів додатка	2	-
9	Ручне тестування ендпоінтів додатка із Persistence Layer	2	-
10	Ручне тестування ендпоінтів додатка із Persistence Layer та DTO	2	-
11	Доповнення доступу до сервісу програмою клієнтом	2	-
12	Використання контейнеру для розміщення сервісу для тестування	2	-
<b>Усього годин</b>		<b>24</b>	<b>4</b>

## 8. Самостійна робота

### 8.1. Види самостійної роботи

№	Вид самостійної роботи	Години (денна/ заочна)	Терміни виконання (денна/ заочна)	Форма та метод контролю (денна/заочна)
1	Опрацювання питань, що виносяться на самостійне вивчення	52/76	Щотижнево	Усне та письмове опитування /Тестування
2	Підготовка до лекційних та практичних занять (робота з інформаційними джерелами: опрацювання першоджерел)	18/26	щотижнево	Усне опитування
3	Підготовка самостійних питань з тематики дисципліни (опрацювання теоретичних основ прослуханого лекційного матеріалу)	12/16	щотижнево	Усне та письмове опитування
4	Індивідуальні творчі завдання (вирішення і письмове оформлення завдань, схем, діаграм, інших робіт графічного характеру; презентації за заданою проблемною тематикою, дослідницькі проекти)	12/14	щотижнево	Спостереження за виконанням, обговорення, виступ з презентацією
5	Підготовка до контрольних робіт та тестування (самостійне опрацювання тестів відповідно до теми практичного заняття; самостійне розв'язання типових задач, ситуаційних вправ)	6/8	1 раз на 2 тижні	Тестування у системі АСУ СОКРАТ
<b>Разом</b>		<b>100/140</b>		

### Перелік питань для самостійного опрацювання (денна форма навчання)

№ п/п	Назва теми	Кількість годин
1	<b>Тема 1. Вступ до тестування програмного забезпечення.</b> Важливість тестування програмного забезпечення для забезпечення надійності. Роль тестування програмного забезпечення в підвищенні задоволеності користувачів. Реальні приклади збоїв програмного забезпечення через недостатнє тестування.	4
2	<b>Тема 2. Основи тестування.</b> Розуміння етапу планування тесту. Розробка тестів і документація. Значення простежуваності в тестуванні.	4
3	<b>Тема 3. Види тестування.</b> Поглиблене дослідження модульного тестування та його переваг. Стратегії і найкращі практики інтеграційного тестування. Роль тестування системи в забезпеченні повної функціональності.	4
4	<b>Тема 4. Автоматизація тестування за допомогою Selenium.</b> Знайомство з Selenium WebDriver і його можливостями. Створення сценаріїв автоматизованих тестових випадків з Selenium. Обробка динамічних веб-елементів і стратегій очікування в Selenium.	4
5	<b>Тема 5. Методи розробки тестів.</b> Поділ еквівалентності. Розбиття вхідних даних на класи еквівалентності. Аналіз граничних значень. Тестування на краях вхідних доменів. Тестування таблиці рішень. керування складними комбінаціями умов.	4

6	<b>Тема 6. Виконання тесту та звітність.</b> Впровадження наборів тестів для регресійного тестування. Ефективне повідомлення про дефекти за допомогою систем відстеження проблем. Методи пріоритезації тестових випадків.	4
7	<b>Тема 7. Безперервне тестування в хмарних середовищах.</b> Налаштування хмарної інфраструктури тестування. Безперервна інтеграція та конвеєрна інтеграція безперервного розгортання (CI/CD). Організація тестів у хмарних середовищах для масштабованості.	4
8	<b>Тема 8. Тестування безпеки та контейнери Docker.</b> Виявлення типових вразливостей безпеки в контейнерних програмах. Інструменти та техніки для сканування образів Docker на наявність вразливостей. Написання тестів безпеки для контейнерних програм.	4
9	<b>Тема 9. Тестування продуктивності в Kubernetes.</b> Планування та проектування тестів ефективності. Використання Kubernetes для створення реалістичних тестових середовищ. Аналіз та інтерпретація результатів тестування продуктивності.	4
10	<b>Тема 10. Перевірка зручності використання та взаємодії з користувачем.</b> Проведення інтерв'ю та опитувань користувачів для тестування зручності використання. Оцінка доступності та відповідності стандартам юзабіліті. Збір і аналіз відгуків користувачів для покращення UX.	4
11	<b>Тема 11. Тестування в середовищах розробки.</b> Налаштування робочих процесів тестування середовища розробки. Найкращі методи перевірки коду для раннього виявлення дефектів. Стратегії впровадження та виконання тестів, орієнтованих на розробника.	4
12	<b>Тема 12. Тестування прийнятності користувача (UAT) у виробництві.</b> Залучення кінцевих користувачів до планування та виконання UAT. Визначення загальних проблем UAT та їх вирішення. Забезпечення плавного переходу від UAT до виробництва.	4
13	<b>Тема 13. Сучасні технології постійного вдосконалення тестування.</b> Впровадження циклів зворотного зв'язку для постійного вдосконалення. Використання показників тестування та показників ефективності. Заохочення культури навчання та адаптації в командах тестування.	4
<b>Усього годин</b>		<b>52</b>

**Перелік питань для самостійного опрацювання  
(заочна форма навчання)**

№ п/п	Назва теми	Кількість годин
1	<b>Тема 1. Вступ до тестування програмного забезпечення.</b> Важливість тестування програмного забезпечення для забезпечення надійності. Роль тестування програмного забезпечення в підвищенні задоволеності користувачів. Реальні приклади збоїв програмного забезпечення через недостатнє тестування.	6
2	<b>Тема 2. Основи тестування.</b> Розуміння етапу планування тесту. Розробка тестів і документація. Значення простежуваності в тестуванні.	6
3	<b>Тема 3. Види тестування.</b> Поглиблене дослідження модульного тестування та його переваг. Стратегії і найкращі практики інтеграційного тестування. Роль тестування системи в забезпеченні повної функціональності.	6

4	<b>Тема 4. Автоматизація тестування за допомогою Selenium.</b> Знайомство з Selenium WebDriver і його можливостями. Створення сценаріїв автоматизованих тестових випадків з Selenium. Обробка динамічних веб-елементів і стратегій очікування в Selenium.	6
5	<b>Тема 5. Методи розробки тестів.</b> Поділ еквівалентності. Розбиття вхідних даних на класи еквівалентності. Аналіз граничних значень. Тестування на краях вхідних доменів. Тестування таблиці рішень. керування складними комбінаціями умов.	6
6	<b>Тема 6. Виконання тесту та звітність.</b> Впровадження наборів тестів для регресійного тестування. Ефективне повідомлення про дефекти за допомогою систем відстеження проблем. Методи пріоритизації тестових випадків.	6
7	<b>Тема 7. Безперервне тестування в хмарних середовищах.</b> Налаштування хмарної інфраструктури тестування. Безперервна інтеграція та конвеєрна інтеграція безперервного розгортання (CI/CD). Організація тестів у хмарних середовищах для масштабованості.	6
8	<b>Тема 8. Тестування безпеки та контейнери Docker.</b> Виявлення типових вразливостей безпеки в контейнерних програмах. Інструменти та техніки для сканування образів Docker на наявність вразливостей. Написання тестів безпеки для контейнерних програм.	6
9	<b>Тема 9. Тестування продуктивності в Kubernetes.</b> Планування та проектування тестів ефективності. Використання Kubernetes для створення реалістичних тестових середовищ. Аналіз та інтерпретація результатів тестування продуктивності.	6
10	<b>Тема 10. Перевірка зручності використання та взаємодії з користувачем.</b> Проведення інтерв'ю та опитувань користувачів для тестування зручності використання. Оцінка доступності та відповідності стандартам юзабіліті. Збір і аналіз відгуків користувачів для покращення UX.	6
11	<b>Тема 11. Тестування в середовищах розробки.</b> Налаштування робочих процесів тестування середовища розробки. Найкращі методи перевірки коду для раннього виявлення дефектів. Стратегії впровадження та виконання тестів, орієнтованих на розробника.	6
12	<b>Тема 12. Тестування прийнятності користувача (UAT) у виробництві.</b> Залучення кінцевих користувачів до планування та виконання UAT. Визначення загальних проблем UAT та їх вирішення. Забезпечення плавного переходу від UAT до виробництва.	5
13	<b>Тема 13. Сучасні технології постійного вдосконалення тестування.</b> Впровадження циклів зворотного зв'язку для постійного вдосконалення. Використання показників тестування та показників ефективності. Заохочення культури навчання та адаптації в командах тестування.	5
<b>Усього годин</b>		<b>76</b>

#### 8.4. Орієнтовний перелік тем індивідуальних творчих завдань

##### Тема 1. Вступ до тестування програмного забезпечення.

1. Вступ до тестування програмного забезпечення: основні концепції та підходи.
2. Важливість тестування для забезпечення надійності програмного

забезпечення.

3. Роль тестування в підвищенні задоволеності користувачів та зниженні кількості дефектів.

4. Приклади збоїв програмного забезпечення через недостатнє або відсутнє тестування (наведення реальних кейсів).

5. Основні типи тестування (функціональне, нефункціональне, регресійне).

6. Автоматизація тестування: інструменти та переваги.

7. Взаємодія між командами розробки та тестування в процесі забезпечення якості.

8. Розробка тестових сценаріїв та аналіз результатів тестування.

9. Етичні аспекти тестування програмного забезпечення: чи завжди все треба тестувати?

## **Тема 2. Основи тестування.**

1. Основи тестування програмного забезпечення: ключові принципи та підходи.

2. Етап планування тестування: що включає і чому важливий.

3. Створення тестових сценаріїв: методології, критерії та практичні приклади.

4. Документування результатів тестування: важливість, формати та приклади.

5. Роль простежуваності в тестуванні: що це таке і чому важливо відслідковувати зв'язок між вимогами, тестами та дефектами.

6. Види тестування та їх застосування на різних етапах розробки ПЗ.

7. Автоматизація тестів: створення ефективних автоматизованих тестів та їх документація.

8. Оцінка покриття тестування: способи вимірювання та використання.

9. Інструменти для відстеження та керування тестуванням: огляд популярних рішень.

## **Тема 3. Види тестування.**

1. Види тестування програмного забезпечення: огляд та порівняння.

2. Модульне тестування: поглиблене дослідження, переваги та приклади.

3. Стратегії інтеграційного тестування: як забезпечити якісну інтеграцію компонентів.

4. Кращі практики інтеграційного тестування: шаблони та підходи для підвищення ефективності.

5. Роль системного тестування: забезпечення повної функціональності програмного продукту.

6. Тестування на регресію: важливість повторної перевірки після змін.

7. Нефункціональне тестування: тестування продуктивності, безпеки та зручності використання.

8. Тестування на рівні приймання: перевірка відповідності кінцевим

вимогам замовника.

9. Ручне vs. автоматизоване тестування: коли і яке застосовувати.

#### **Тема 4. Автоматизація тестування за допомогою Selenium.**

1. Вступ до автоматизації тестування: огляд інструментів та підходів.
2. Знайомство з Selenium WebDriver: можливості та архітектура.
3. Створення базових сценаріїв автоматизованого тестування за допомогою Selenium.
4. Обробка динамічних веб-елементів в Selenium: стратегії та приклади.
5. Стратегії очікування в Selenium: використання явних та неявних очікувань для підвищення надійності тестів.
6. Робота з формами та елементами керування: автоматизація заповнення форм та перевірка даних.
7. Інтеграція Selenium з інструментами тестування (JUnit, TestNG) для створення структурованих тестових наборів.
8. Тестування веб-додатків з використанням різних браузерів за допомогою Selenium.
9. Кращі практики автоматизації тестування за допомогою Selenium: оптимізація продуктивності та підтримка тестів.

#### **Тема 5. Методи розробки тестів.**

1. Вступ до методів розробки тестів: огляд теорій і підходів.
2. Поділ еквівалентності: теорія та приклади використання для ефективного тестування.
3. Розбиття вхідних даних на класи еквівалентності: як уникнути надмірного дублювання тестів.
4. Аналіз граничних значень: як правильно тестувати крайні випадки введення даних.
5. Тестування на краях вхідних доменів: приклади граничних ситуацій та їх обробка.
6. Тестування таблиці рішень: інструменти для аналізу складних логічних умов і дій.
7. Методи керування складними комбінаціями умов: підходи до оптимізації тестування.
8. Застосування технік поділу еквівалентності та аналізу граничних значень на практиці.
9. Переваги і обмеження кожної методики тестування: порівняльний аналіз.

#### **Тема 6. Виконання тесту та звітність.**

1. Виконання тестів: підходи та найкращі практики.
2. Впровадження наборів тестів для регресійного тестування: створення та підтримка тестових наборів.
3. Як ефективно повідомляти про дефекти за допомогою систем відстеження проблем (Jira, Bugzilla).

4. Методи пріоритезації тестових випадків: визначення важливості тестів для різних аспектів продукту.
5. Оцінка ефективності тестування: показники успіху та звітність.
6. Автоматизація регресійного тестування: як скоротити час перевірки при збереженні якості.
7. Управління результатами тестування: аналіз, документування та відстеження прогресу.
8. Вплив пріоритезації на швидкість випуску продукту: як правильно вибрати критичні тестові випадки.
9. Інтеграція систем звітності з процесами розробки: відстеження статусу виправлення дефектів.

### **Тема 7. Безперервне тестування в хмарних середовищах.**

1. Безперервне тестування в хмарних середовищах: огляд концепції та переваги.
2. Налаштування хмарної інфраструктури тестування: як забезпечити ефективність та надійність.
3. Інтеграція безперервного тестування з CI/CD конвеєром: автоматизація розгортання та тестування.
4. Масштабованість тестування у хмарі: як організувати тести для великих і складних систем.
5. Автоматизація тестів у хмарних середовищах: приклади практичної реалізації.
6. Стратегії зниження витрат на тестування в хмарі: як оптимізувати використання ресурсів.
7. Моніторинг та аналіз тестових результатів у хмарних середовищах: інструменти та методи.
8. Використання контейнерних технологій для масштабованого тестування у хмарі (Docker, Kubernetes).
9. Безпека під час тестування у хмарних середовищах: ризики та підходи до захисту даних.

### **Тема 8. Тестування безпеки та контейнери Docker.**

1. Типові вразливості безпеки у контейнерних програмах: аналіз і приклади.
2. Інструменти для сканування образів Docker на наявність вразливостей: огляд та використання.
3. Як виявити вразливості в контейнеризованих додатках: практичні підходи та інструменти.
4. Написання тестів безпеки для контейнерних програм: кроки та приклади.
5. Автоматизація безпекових тестів для Docker-контейнерів: інтеграція з CI/CD процесами.
6. Контроль версій і моніторинг безпеки образів Docker: як уникнути вразливостей.



7. Використання Snyk, Clair або Trivy для сканування контейнерів на вразливості.

8. Найкращі практики захисту контейнерів Docker: як мінімізувати ризику.

9. Тестування мережевої безпеки в контейнерних середовищах: забезпечення ізоляції та захисту даних.

### **Тема 9. Тестування продуктивності в Kubernetes.**

1. Основи тестування продуктивності в Kubernetes: чому це важливо для сучасних додатків.

2. Планування та проектування тестів ефективності в контейнеризованих середовищах.

3. Використання Kubernetes для створення реалістичних тестових середовищ: кроки та найкращі практики.

4. Інструменти для тестування продуктивності в Kubernetes (наприклад, JMeter, k6, Locust).

5. Тестування під навантаженням в Kubernetes: симуляція реальних користувацьких сценаріїв.

6. Автоматизація тестування продуктивності в CI/CD конвеєрі з використанням Kubernetes.

7. Як аналізувати та інтерпретувати результати тестування продуктивності в Kubernetes.

8. Масштабованість тестових середовищ у Kubernetes: стратегії та підходи.

9. Оптимізація продуктивності контейнеризованих додатків на основі результатів тестування.

### **Тема 10. Перевірка зручності використання та взаємодії з користувачем.**

1. Проведення інтерв'ю та опитувань користувачів для оцінки зручності використання: методології та приклади.

2. Оцінка доступності веб-додатків: відповідність стандартам WCAG та іншим нормам.

3. Виявлення проблем з юзабіліті через тестування реальних користувацьких сценаріїв.

4. Збір та аналіз відгуків користувачів для покращення UX: підходи та техніки.

5. Використання юзабіліті-тестування для підвищення задоволеності користувачів продуктом.

6. Тестування доступності для людей з обмеженими можливостями: важливість інклюзивного дизайну.

7. Інструменти для оцінки зручності використання та відстеження взаємодії з користувачем (наприклад, Hotjar, Google Analytics).

8. Найкращі практики для створення ефективного UX-дизайну на основі зворотного зв'язку.

9. Інтерпретація результатів юзабіліті-тестування: як зробити висновки для покращення продукту.

### **Тема 11. Тестування в середовищах розробки.**

1. Налаштування робочих процесів тестування в середовищах розробки: автоматизація та інтеграція.

2. Найкращі методи перевірки коду для раннього виявлення дефектів: практики та інструменти.

3. Використання статичного аналізу коду для виявлення дефектів на ранніх стадіях розробки.

4. Стратегії впровадження тестів, орієнтованих на розробника: TDD (Test-Driven Development) і BDD (Behavior-Driven Development).

5. Роль модульного тестування в середовищах розробки: автоматизація і перевірка на рівні модулів.

6. Інтеграційне тестування в процесі розробки: виявлення проблем взаємодії між компонентами.

7. Використання хмарних середовищ для тестування в процесі розробки: переваги і виклики.

8. Автоматизація CI/CD конвеєрів для безперервного тестування в процесі розробки.

9. Як знизити ризик дефектів на продакшн-середовищі через ефективне тестування під час розробки.

### **Тема 12. Тестування прийнятності користувача (UAT) у виробництві.**

1. Залучення кінцевих користувачів до планування та виконання UAT: методи та практичні підходи.

2. Визначення ролей та відповідальності у процесі UAT: участь користувачів та розробників.

3. Типові проблеми, що виникають під час UAT, та шляхи їх вирішення.

4. Створення ефективних тестових сценаріїв для UAT: як забезпечити покриття всіх бізнес-вимог.

5. Впровадження UAT у процесі CI/CD: автоматизація та інтеграція з виробничим середовищем.

6. Забезпечення плавного переходу від UAT до виробництва: мінімізація ризиків під час розгортання.

7. Збір та аналіз зворотного зв'язку користувачів після UAT: як враховувати відгуки у майбутніх релізах.

8. Роль UAT у забезпеченні відповідності продукту бізнес-вимогам і очікуванням користувачів.

9. Як уникнути типових помилок під час планування та виконання UAT.

### **Тема 13. Сучасні технології постійного вдосконалення тестування.**

1. Впровадження циклів зворотного зв'язку для постійного вдосконалення процесів тестування: найкращі практики.
2. Використання показників тестування та ключових показників ефективності (КРІ) для моніторингу якості тестування.
3. Аналіз та оптимізація процесів тестування на основі метрик.
4. Створення та підтримка культури безперервного навчання в командах тестування.
5. Використання інструментів автоматизованого тестування для постійного вдосконалення тестової інфраструктури.
6. Як інтегрувати постійне вдосконалення в CI/CD цикли тестування.
7. Оцінка ефективності процесів тестування та впровадження змін на основі даних.
8. Як впроваджувати нові технології тестування без порушення існуючих процесів.
9. Заохочення культури експериментів та адаптації в командах тестування: підходи та приклади.

## 9. Методи викладання та демонстрування результатів навчання

Лекція (в т.ч. використання мультимедійних лекцій)

Студентські презентації та виступи на наукових заходах

Презентації результатів виконаних завдань та досліджень

Практичні роботи

## 10. Форми поточного та підсумкового контролю

Іспит

Тестування

Захист практичної роботи

Самоконтроль

Самопрезентації

Захист практичної роботи

Презентація результатів дослідження

## 11. Критерії оцінювання результатів навчання

### 11.1. Розподіл балів за видами навчальної діяльності

	Вид навчальної діяльності	Бали	
		денна форма навчання	заочна форма навчання
<b>Атестація 1</b>			
1	Участь у дискусіях на лекційних заняттях	6	4
2	Участь у роботі на практичних заняттях	18	3
3	Виконання контрольних робіт / тестування	3	14
4	Самостійна робота (підготовка питань, винесених на самостійне опрацювання, індивідуальні завдання)	3	14
	<b>Всього за атестацію 1</b>	<b>30</b>	<b>35</b>
<b>Атестація 2</b>			
5	Участь у дискусіях на лекційних заняттях	7	2
6	Участь у роботі на практичних заняттях	18	3
7	Виконання контрольних робіт / тестування	3	15
8	Самостійна робота (підготовка питань, винесених на самостійне опрацювання, індивідуальні завдання)	2	15
	<b>Всього за атестацію 2</b>	<b>30</b>	<b>35</b>
<b>Разом</b>		<b>60</b>	<b>70</b>
<b>Показники наукової, інноваційної, навчальної, виховної роботи та студентської активності</b>		<b>10</b>	<b>-</b>
<b>Підсумкове тестування</b>		<b>30</b>	<b>30</b>
<b>Разом</b>		<b>100</b>	<b>100</b>

## 11.2. Відповідність шкал оцінок якості засвоєння навчального матеріалу

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	A	відмінно
82-89	B	добре
75-81	C	
66-74	D	задовільно
60-65	E	
35-59	FX	незадовільно з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

Якщо здобувач упродовж семестру за підсумками контрольних заходів набрав (отримав) менше половини максимальної оцінки з навчальної дисципліни (менше 35 балів), то він не допускається до іспиту.

## 11.3. Критерії поточного оцінювання знань здобувачів вищої освіти

Участь у дискусіях на лекційних та практичних заняттях, виконання контрольних робіт, індивідуальні та групові творчі завдання, тестування	Критерії оцінювання
90 – 100	В повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час усних виступів та письмових відповідей, глибоко та всебічно розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому обов'язкову та додаткову літературу. Правильно вирішив усі тестові завдання.
75-89	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає під час усних виступів та письмових відповідей, в основному розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому обов'язкову літературу. Але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки. Правильно вирішив більшість тестових завдань.
60-74	В цілому володіє навчальним матеріалом викладає його основний зміст під час усних виступів та письмових відповідей, але без глибокого всебічного аналізу, обґрунтування та аргументації, без використання необхідної літератури допускаючи при цьому окремі суттєві неточності та помилки. Правильно вирішив половину тестових завдань.
35-59	Не в повному обсязі володіє навчальним матеріалом. Фрагментарно, поверхово (без аргументації та

	обґрунтування) викладає його під час усних виступів та письмових відповідей, недостатньо розкриває зміст теоретичних питань та практичних завдань, допускаючи при цьому суттєві неточності, правильно вирішив меншість тестових завдань.
16-34	Частково володіє навчальним матеріалом не в змозі викласти зміст більшості питань теми під час усних виступів та письмових відповідей, допускаючи при цьому суттєві помилки. Правильно вирішив окремі тестові завдання.
0-15	Не володіє навчальним матеріалом та не в змозі його викласти, не розуміє змісту теоретичних питань та практичних завдань. Не вирішив жодного тестового завдання.

## 12. Методичне забезпечення

Суприган В.А. Методичні вказівки до виконання практичних робіт з дисципліни «Тестування програмного забезпечення» – «план. 2024»

## 13. Рекомендовані джерела інформації

### Основні

1. Хмарні технології: навч. посіб. / [О.В. Зінченко, С.М. Іщеряков, С.В. Прокопов, С.О. Сєрих, В.В. Василенко]. – К: ФОП Гуляєва В. М., 2020. – 74 с.
2. Операційні системи: навч. посіб. / [І. М. Федотова-Півень, І. В. Миронець, О. Б. Півень, С. В. Сисоєнко, Т. В. Миронюк]; за ред. В. М. Рудницького. – Харків: ТОВ «ДІСА ПЛЮС», 2019. – 216 с.
3. Gayathri Mohan Full Stack Testing. A Practical Guide for Delivering High Quality Software. O'reilly Media, 2022. – 406 p.
4. Saleem Siddiqui Learning Test-Driven Development: A Polyglot Guide to Writing Uncluttered Code 1st Edition. O'reilly Media, 2021. – 280 p.
5. Matthew Heusser, Michael Larsen Software Testing Strategies: A testing guide for the 2020s. Packt Publishing, 2023. – 378 p.
6. Mauricio Aniche Effective Software Testing: A developer's guide. Manning, 2022. – 328 p.

### Додаткові

1. Brian Okken Python Testing with pytest. Simple, Rapid, Effective, and Scalable. 2nd Edition. Pragmatic Bookshelf, 2022. – 274 p.
2. Alexander Tarlinder Developer Testing: Building Quality into Software (Addison-Wesley Signature Series (Cohn)) 1st Edition. Addison-Wesley Professional, 2017. – 314 p.
3. Andrew Pollner, Mark Fewster, Ina Schieferdecker Test Automation Engineer: Guide to the ISTQB Advanced Level Certification. Rocky Nook, 2018. – 300 p.
4. Shen J. J. Software Testing. Techniques, Principles, and Practices. Paperback. Independently published, 2019. – 286 p.
5. Rex Black, Dorothy Graham, Erik van Veenendaal Foundations of Software Testing ISTQB Certification, 4th edition. Cengage Learning EMEA, 2020. – 288 p.
6. Lucas da Costa Testing JavaScript Applications. Manning Publications Co., 2021. – 512 p.
7. Brian Okken Python Testing with pytest. Simple, Rapid, Effective, and Scalable. 2nd Edition. Pragmatic Bookshelf, 2022. – 274 p.
8. Kristin Jackvony The Complete Software Tester: Concepts, Skills, and Strategies for High-Quality Testing. Kindle Edition, 2021. – 514 p.
9. Nikolina Finska Modern Game Testing: Learn how to test games like a pro, optimize testing effort, and skyrocket your QA career. Packt Publishing, 2023. – 232 p.

10. Vladimir Khorikov Unit Testing Principles, Practices, and Patterns: Effective testing styles, patterns, and reliable automation for unit testing, mocking, and integration testing with examples in C#. Manning, 2020. – 304 p.

#### **Інтернет-ресурси:**

1. Офіційний сайт компанії JetBrains [Електронний ресурс] / JetBrains. Режим доступу: <https://www.jetbrains.com>
2. Офіційний сайт Верховної Ради України [Електронний ресурс] / Верховна Рада України. Режим доступу: <http://www.rada.gov.ua>
3. Офіційний сайт Кабінету Міністрів України [Електронний ресурс] / Кабінет Міністрів України. Режим доступу: <http://www.kmu.gov.ua>
4. Офіційний сайт Державної служби статистики України [Електронний ресурс] / Державна служба статистики України. Режим доступу: <http://www.ukrstat.gov.ua>
5. Офіційний сайт Національного банку України [Електронний ресурс] / Національний банк України. Режим доступу: <http://www.bank.gov.ua>
6. Офіційний сайт Національної комісії з цінних паперів та фондового ринку [Електронний ресурс] / Національна комісія з цінних паперів та фондового ринку. Режим доступу: <http://www.nssmc.gov.ua>
7. Офіційний сайт Міністерства фінансів [Електронний ресурс] / Міністерство фінансів. Режим доступу: <http://www.minfint.gov.ua>
8. Офіційний сайт Світового банку [Електронний ресурс] / The World Bank. Режим доступу: <https://www.worldbank.org/uk/country/ukraine>
9. Офіційний сайт Державної регуляторної служби України [Електронний ресурс] / Державна регуляторна служба України. Режим доступу: <https://www.drs.gov.ua>
10. Офіційний сайт. Нормативні акти України [Електронний ресурс] / Правові системи. Режим доступу: <http://www.nau.ua>